Administrative Unit: Computer and Mathematical Sciences Department

Course Prefix and Number: CISS 240

Course Title: Introduction to Programming

Number of: Credit Hours: 4  Lecture Hours: 3  Laboratory Hours: 1

Catalog Description: Program design and development using C++. A disciplined approach to problem solving and algorithm development will be stressed using top-down design. Topics include syntax and semantics, input/output, selection, iterative constructs, functions, data types, arrays, strings, pointers, and recursion. Prerequisites: MATH 150, CISS 170.

Prerequisite(s)/Corequisite(s): MATH 150, CISS 170.

Text(s): Most current edition of the following:

Dietel, H. M. & Dietel, P. J.  C++ How to Program. Prentice Hall.

Course Objectives:

• To understand procedural C++ programming.
• To decompose simple problems and map decomposition onto C++ language syntax and semantics.
• To write thoroughly documented programs.
• To write well-tested and appropriately robust programs.

Measurable Learning Outcomes:

• Sample questions (Q) and exercises (E) are provided in order to indicate the approximate level of competence expected for each outcome.
• Write program documentation that is succinct, thorough, and maintains standards of good English.
• Compile, load and run a C++ program using a compiler.
• Explain what the assembly process does.
• Explain what a compiler does.
• Explain object code linking.
• Explain program loading.
• Compose and evaluate mathematical expressions in C++. E: Evaluate the following expression x = 24 / 8%(3*7/2 + 4/6);
• Use basic math functions from compiler-provided libraries.
• Use appropriate data types:
Q: What C++ data type would be most appropriate for each of the following:
• Processing wage and salary information,
• Storing the number of dependents a person has, and
• Calculating the interest on the national debt.
Q: Give two reasons why a programmer might choose to use a data type of unsigned integer rather than signed integer.

Q: Explain why a programmer would use type “float” rather than type “double”.

- Demonstrate when to use and how to write while loops.

E: Write a program that will read integers until a negative value is read, then print the sum and average of the values read, excluding the final negative value.

- Demonstrate when to use and how to write for loops.

E: Write a loop that will sum all numbers between 1 and 10,000 that are divisible by 6.

Q: Is the following syntax legal, and if yes, what will it do?

E: Write a program that will read integers until a negative value is read, then print the sum and average of the values read, excluding the final negative value.

- Demonstrate when to use and how to write do-while loops.

Q: What is the difference between a “while” logo and a “do-while” loop?

E: Write a “do-while” loop that will print the elements on the main diagonal of a two-dimensional square array up to and including the first negative value found or the last element on the main diagonal if no negative values are found.

- Demonstrate when to use and how to write switch statements.

E: Write a switch statement that will read characters from the keyboard and count all upper-case characters, all lower-case letters, all write-space characters (space, tab, newline), and all other characters.

- Demonstrate when to use and how to write if-else statements.

E: Write a series of nested “if-else” statements to sort three integer values.

- Demonstrate when to use and how to write compound Boolean expression using Boolean operators.

E: Write an “if” statement that will discriminate between values that are divisible by 2, or 9, or both 11 and 15 and those that are divisible by 7, 33, or all of 19, 31, 41, and 71 and those that are not divisible by either 131 or 133.

E: Write a series of “if” statements, one or more, that will print a letter grade of “A”, “B”, …, “F” based on a value in a variable, “score”, as follows:

- Demonstrate how to call and construct functions.

- Demonstrate how to use scope rules.

```c++
int a  = 0, b  = 1;
int demo (  int,  int &  );
void main ( )
{int a  = 3,  b  = 4;
demo (a,b);
cout << a << b << end1;
demo (b,a);
cout << a << b << end1;
a  =  b+1;
```
cout << a << b << end1;
}  

void demo ( int x, int &y)  
{  
int a = 4;  
++y;  
x = a + b + y;  
y = a + b + c;  
}  

- Demonstrate and use formal and actual parameters.  
Q: What is the difference between formal and actual parameters?  
- Demonstrate how to use value and reference parameters.  
- Demonstrate how to use and construct template functions.  
E: Write function prototypes for each of the following:  
  - given the radius, compute the area of a circle  
  - given the radius, compute the area and circumference of a circle  
  - interchange (swap) two values  
  - find and return the maximum of 3 values  
E: Write a function template that will find the largest value from four values. Give an example of how the function would be called using four float variables, using four integer variables.  
- Demonstrate how to use and construct overloaded functions.  
- Demonstrate how to use and write function prototypes.  
- Declare and initialize one- and two-dimensional arrays.  
- Pass arrays as parameters to function.  
Write a function that will accept a two-dimensional double array parameter and print the values on the upper-right to lower-left diagonal.  
- Write a function that will sort an array of characters.  
- Write a template for a function that will sort an array into ascending order. Write a driver program that uses the function to sort an integer, float, and char array.  
- Write recursive functions.  
E: Write a recursive function that will add the elements in an array.  
- Trace flow of control of recursive functions.  
Q: What does the following function do?  
int mystery ( int a, int b )  
{  
if ( b < 0 )  
cout << “error” << end1;  
else if ( b == 0 )  
return 1;  
else  
return b * mystery ( a, b - 1 );  
}
Topical Outline (major areas of coverage):

- Basic components of a computer
- Compiling, assembling, linking and loading a program
- History of C++
- Variables
- Assignment statements
- if statements
- Repetition statements: while, for, do/while
- Multiple selection with switch statements
- Break and continue statements
- Functions
- Scope rules
- Recursion
- Reference parameters
- Function templates
- Arrays
- Array parameters
- Sorting arrays
- Pointers and strings
- Pointers and arrays

Recommended maximum class size for this course: 20

Library Resources: Online databases are available at http://www.ccis.edu/offices/library/resources.asp. You may access them from off-campus using your eServices login and password when prompted.

Prepared by: Yihsiang Liow

Name ____________________________ Signature ____________________________

Date: March 20, 2006

NOTE: The intention of the master syllabus is to provide an outline of the contents of this course, as specified by the faculty of Columbia College, regardless of who teaches the course, when it is taught or where it is taught. Faculty members teaching this course for Columbia College are expected to facilitate learning pursuant to the course objectives and cover the subjects listed in the topical outline. However, instructors are also encouraged to cover additional topics of interest so long as those topics are relevant to the course’s subject. The master syllabus is, therefore, prescriptive in nature but also allows for a diversity of individual approaches to course material.